UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/685,654	10/14/2003	Jeffrey Rupley II	42P17014	5664

8791 7590 10/31/2006

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
LAI, VINCENT	
ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 10/31/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/685,654	**Applicant(s)** RUPLEY ET AL.	
	Examiner Vincent Lai	**Art Unit** 2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 October 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-53* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21 and 23-53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

FRITZ FLEMING
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

10/27/2006

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *8/1/2006*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-53 have been examined.

Response to Amendment

2. Acknowledgment is made of the amendments to the specification, claims, and title.

3. Objection to the specification, title, and claims are withdrawn after considering amendments.

Response to Arguments

4. Applicant's arguments filed 1 August 2006 have been fully considered but they are not persuasive.

All arguments on new limitations are addressed below in the rejections.

Applicant argues, "Morrison fails to disclose, teach or suggest all the claim limitations of Claim 30. For example, amended Claim 30 recites, in part, "a micro-operation generator to receive a predicated instruction, the predicated instruction being one of a plurality of instructions, **the plurality including arithmetic instructions and memory instructions**." (Claim 30, in part, emphasis added). Such limitation is not disclosed, taught, or suggested by Morrison. The Examiner claims that this limitation is inherent in the teaching of the Morrison. However, Applicants disagree."

Applicant had not argued how a plurality of instructions does not encompass arithmetic and memory instructions. One having ordinary skill in the art would recognize that arithmetic and memory instructions are commonly used and disclosure of arithmetic and memory instructions would not be novel. Figure 3 explicitly shows an ADD operation, which is an arithmetic operation. In modern computer systems, data must be saved into some form of memory or computations will be lost and therefore be of no use therefore memory instructions exist. Furthermore, Morrison alludes to memory instructions when discussing operation of system. Referring to column 4, lines 23-27, Morrison teaches storing values into predicate registers.

Specification

5. The disclosure is objected to because of the following informalities: Amended paragraph [00011] starts with "Fig.. 3". It is suggested that it be changed to "Fig. 3" to consistent with the rest of the specification.

Appropriate correction is required.

Claim Objections

6. Claim 35 is objected to because of the following informalities: "Predicated" is spelled incorrectly. Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

7. Claims 1-29 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Evidence that destination registers must be distinct could not be found in the specification.

[Examiner's Note: For future references, substantial amendments that change scope of claims should be supported by citation of the specification.]

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1-6, 19-22, 24-26, 28, 30-40, and 53 are rejected under 35 U.S.C. 102(b) as being anticipated by Morrison (US Patent No. 6,170,052).

Regarding **independent claim 1**, Morrison discloses *an apparatus comprising: a micro-operation generator* [see Morrison, Fig. 4, elements 54,56; Col. 5, lines 20-22] *to receive a predicated instruction that indicates a plurality of destination registers* [Examiner's note: Morrison discloses decoders that function to translate a macro-operation into micro-operations. Morrison does not explicitly state an instruction containing multiple destination registers. However, at the time of invention, VLIW (macro-instruction) processors were a well known type of processor, and a fundamental feature of VLIW processors is the ability to receive a VLIW that contains multiple, smaller instructions, known as micro-operations. With reference to the included article from the Microsoft Computer Dictionary, a VLIW is clearly an instruction that contains multiple destination registers ("An architecture that combines many simple instructions into a single long instruction word that uses different registers."), thus, the macroinstructions disclosed by Morrison clearly contain multiple micro-operations containing a single destination register (as shown by the micro-operations in Fig. 3, element 35). Since a macro-instruction contains at least two micro-instructions, it is clear that multiple destination registers exist within a macro-instruction disclosed by Morrison.]*; the micro-operation generator further to generate a plurality of micro-operations that implement the instruction* [see Morrison, Col. 5, lines 20-27]*; the micro-operation generator further to decompose the instruction into the plurality of micro-operations such that each of the micro-operations indicates a single destination register* [see Morrison, Fig. 3, elements 31-33; Fig. 4, elements 37-39; Examiner's note: it is clear from the figures cited that the generated micro-ops contain a single destination register (Z). Furthermore, it is clear that the micro-op disclosed by Morrison would have been generated from the macro-operations disclosed by Morrison, thus multiple single destination micro-operations would have been stored in a macro-operation creating a multiple destination macro-operation.] that is distinct form the destination registers indicated by other micro-operations [see Morrison, Fig. 5, elements 58-60; Examiner's note: the existence of renaming units mean that each micro-operation will have its own register].

Regarding **claim 2**, Morrison discloses *the apparatus of claim 1, wherein the plurality of micro-operations includes one or more standard micro-operations to provide the basic functionality indicated by the instruction* [see Morrison, Col. 5, lines 22-26; Figs. 3 and 4].

Regarding **claim 3**, Morrison discloses *the apparatus of claim 1, wherein the plurality of micro-operations includes one or more predication-support micro-operations* [see Morrison, Col. 5, lines 22-26; Fig. 3, element 31].

Regarding **claim 4**, Morrison discloses *the apparatus of claim 3, wherein: the one or more predication-support micro-operations includes a conditional move micro-operation for each destination register indicated by the instruction* [see Morrison, Fig. 1, element 14], *where the conditional move micro-operation is to select between a first*

(OP{N,M}) *and second proposed output value* (ADD{C,D}) *based on the value of a qualifying predicate associated with the instruction* (P_x).

Regarding **claim 5**, Morrison discloses *the apparatus of claim 4, wherein: the first proposed output value is the incoming value of the associated destination register* [see Morrison, Fig. 1, element 16]*; and the second proposed output value is a computed value* [see Morrison, Fig. 1, element 14 ("ADD{C,D}" as computed value].

Regarding **claim 6**, Morrison discloses *the apparatus of claim 4, wherein: the first proposed output value is the incoming value of the associated destination register* [see Morrison, Fig. 1, element 16]*; and the second proposed output value is a constant value* [Examiner's note: Morrison discloses an example of a predicated instructions (Figs. 3 and 4), Morrison does not explicitly disclose the use of a constant value as a second proposed output value. However, it is inherent that the figures presented by Morrison are exemplary in nature and the true intent of the invention is to allow for the predication of multiple instructions within a given instruction set in a modern processor. Therefore, it is inherent that a constant value may be utilized as a second proposed output, such as in the case of an unconditional move instruction.].

Regarding **independent claim 19**, Morrison discloses *a system comprising: a memory to store an instruction that indicates a plurality of destination registers* [see Morrison, Col.5, lines 17-19; Examiner's note: By definition a macroinstruction contains multiple microinstructions, inherently containing a plurality of destination registers.]; *and a processor coupled to the memory* [see Morrison, Col. 5, lines 39-42]; *wherein the processor includes a microcode engine to generate, for the instruction, a series of micro-operations,* [see Morrison, Col. 5, lines 20-27] *wherein each of the micro-operations indicates a single distinct* [see Morrison, Fig. 5, elements 58-60; Examiner's note: the existence of renaming units mean that each micro-operation will have its own register] *destination operand and a maximum of two source operands* [see Morrison, Fig. 3, element 31; Examiner's note: In element 31, Morrison discloses an example micro-op which clearly contains one destination (Z) and two sources (C,D)].

Regarding **claim 20**, Morrison discloses *the system of claim 19, wherein: the memory is a DRAM* [Examiner's note: It would have been inherent at the time of invention that DRAM was an industry standard in memories designed to store instructions.].

Regarding **claim 21**, Morrison discloses *the system of claim 19, wherein: the instruction is predicated* [see Morrison, Col. 4, lines 21-22]; *and the micro-code engine is further to generate, for each of the destination registers indicated by the instruction, a micro-operation to conditionally move one of two proposed values into the destination register* [see Morrison, Fig. 1, element 14; Examiner's note: In this cite, Morrison discloses an instruction (P_x = 1; Z = ADD{C,D};) which, based on the value of single bit

P_{x}, determines whether to load the value ADD{C,D} into Z or load the value OP{N,M} into Z as stated in element 16 of said Figure.].

Claim 22 has been cancelled.

Regarding **independent claim 24**, Morrison discloses *a method comprising: receiving an instruction that indicates a plurality of destination registers* [see Morrison, Col.5, lines 17-19; Examiner's note: By definition a macroinstruction contains multiple microinstructions, inherently containing a plurality of destination registers. The Applicant is referred to paragraph 9 of this action for further clarification.]*; and generating a plurality of standard micro-operations for the instruction, wherein each of the standard micro-operations indicates a single destination register* [see Morrison, Fig. 3, element 31; Examiner's note: In element 31, Morrison discloses an example micro-op which clear contains one destination (Z) and two sources (C,D) *that is distinct from the destination registers indicated in of the other standard micro-operations* [see Morrison, Fig. 5, elements 58-60; Examiner's note: the existence of renaming units mean that each micro-operation will have its own register].

Regarding **claim 25**, Morrison discloses *the method of claim 24, further comprising: generating, if the instruction is predicated, a plurality of predication-support micro-operations for the instruction* [see Morrison, Col. 4, lines 20-25].

Regarding **claim 26**, Morrison discloses *the method of claim 25, wherein generating a plurality of predication-support micro-operations further comprises: generating a conditional move micro-operation for each destination register indicated by the instruction* [see Morrison, Fig. 1, element 14].

Regarding **claim 28**, Morrison discloses *the method of claim 26, wherein generating a conditional move operation further comprises: if the instruction does not indicate a known constant as a potential output value, generating a micro-operation that receives a first potential output value as an input* [see Morrison, Fig. 3, element 32] *and also receives a second potential output value as an input* [see Morrison, Fig. 4, element 34], *wherein execution of the micro-operation is to cause a processor to move either the first potential output value or the second potential output value into a destination register* [see Morrison, Fig. 3, element 30; Examiner's note: In this code snippet, Morrison is disclosing the movement of either "OP_1{C,D}" or "ADD{0,Z}" to the register "Z", thus supplying a destination register with two differing non-constant values.].

Regarding **independent claim 30**, Morrison discloses an apparatus comprising: a micro-operation generator to receive a predicated instruction [see Morrison, Col. 5, lines 19-20], the predicated instruction being one of a plurality of instructions [see Morrison, Col.5, lines 17-19], the plurality including arithmetic instructions and memory instructions [Examiner's note: It would have been inherent that at the time a modern processor (Col. 1, lines 14-16) such as one disclosed by Morrison would utilize both arithmetic and memory instructions. Furthermore, Morrison does disclose arithmetic instructions (Fig. 3, element 33), and discloses the use of a memory (Col. 5, lines 17-19).]; the micro-operation generator further to generate a plurality of micro-operations that implement the instruction [see Morrison, Col. 5, lines 20-27].

Regarding **claim 31**, Morrison discloses *the apparatus of claim 30, wherein the predicated instruction is an arithmetic instruction.*

Regarding **claim 32**, Morrison discloses *the apparatus of claim 31, wherein the predicated instruction is an add instruction.*

Regarding **claim 33**, Morrison discloses *the apparatus of claim 31, wherein the predicated instruction is a compare instruction.*

Regarding **claim 34**, Morrison discloses *the apparatus of claim 30, wherein the predicated instruction is a memory instruction.*

Regarding **claim 35**, Morrison discloses *the apparatus of claim 34, wherein the predcicated* [sic] *instruction is a load instruction.*

Regarding **claim 36**, Morrison discloses *the apparatus of claim 34, wherein the predicated instruction is a store instruction.*

Regarding **claims 31-36**, It would have been inherent at the time of invention that a modern processor employed by Morrison would have had the capabilities to perform the instructions claim in claims 31-36.

Regarding **claim 37**, Morrison discloses *the apparatus of claim 30, wherein the plurality of micro-operations includes one or more standard micro-operations to provide the basic functionality indicated by the instruction* [see Morrison, Col. 5, lines 22-26; Figs. 3 and 4].

Regarding **claim 38**, Morrison discloses *the apparatus of claim 30, wherein the plurality of micro-operations includes one or more predication-support micro-operations* [see Morrison, Col. 5, lines 22-26; Fig. 3 elements 31-33; Fig. 4, elements 37-39].

Regarding **claim 39**, Morrison discloses *the apparatus of claim 38, wherein: the one or more predication-support micro-operations includes a conditional move micro-operation* [see Morrison, Fig. 1, element 14] *for each destination register indicated by the instruction, where the conditional move micro-operation is to select between a first* (OP{N,M}) *and second proposed output value* (ADD{C,D}) *based on the value of a qualifying predicate associated with the instruction* (P_X).

Regarding **claim 40**, Morrison discloses *the apparatus of claim 39, wherein: the first proposed output value is the incoming value of the associated destination register* [see Morrison, Fig. 1, element 16]; *and the second proposed output value is a computed value* [see Morrison, Fig. 1, element 14 ("ADD{C,D}" as computed value].

Regarding **claim 53**, Morrison inherently discloses *the apparatus of claim 33, wherein the instruction is a parallel compare instruction.* [Examiner's note: Although Morrison does not expressly disclose the use of a parallel compare instruction, it would have been inherent that Morrison intended to use an instruction set for a modern processor at the time of invention which would have contained an parallel compare instruction.].

Morrison discloses *the apparatus of claim 30, wherein: the micro-operation generator is further to decompose the instruction into the plurality of micro-operations such that each of the micro-operations indicates a single destination register* [see Morrison, Fig. 3, elements 31-33; Fig. 4, elements 37-39].

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 7-15, 27, 33, 41-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morrison in view of Intel ("Intel® Itanium™ Architecture Software Developer's Manual", Volume 1: Application Architecture, Revision 2.0, December 2001).

Regarding **claim 7**, Morrison discloses the limitations as stated in **claim 6**.

Morrison does not explicitly disclose *the instruction* [being] *a compare instruction*.

Intel does disclose *the instruction* [being] *a compare instruction* [see Intel, §4.3.3, line 1].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 8**,

Morrison and Intel disclose the limitations as stated in **claim 7**.

Intel further discloses *the instruction* [being] *an "or" form compare instruction* [see Intel, Table 4-9, Compare Type: OR, Completer: "or"].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 9**, Morrison and Intel disclose the limitations as stated in **claim 7**.

Intel further discloses *the instruction* [being] *an "and" form compare instruction* [see Intel, Table 4-9, Compare Type: AND, Completer: "and"].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 10**, Morrison and Intel disclose the limitations as stated in **claim 7**.

Intel further discloses *the instruction* [being] *an "and"-complement form compare instruction* [see Intel, Table 4-9, Compare Type: AND, Completer: "andcm"].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 11**, Morrison and Intel disclose the limitations as stated in **claim 7**.

Intel further discloses *the instruction* [being] *an "or"-complement form compare instruction* [see Intel, Table 4-9, Compare Type: OR, Completer: "orcm"].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention,

such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 12**, Morrison discloses the limitations as stated in **claim 4**.

Morrison does not explicitly disclose *the instruction* [being] *a parallel compare instruction.*

Intel does disclose *the instruction* [being] *a parallel compare instruction* [see Intel, §4.3.3, lines 13-14].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 13**, Morrison and Intel disclose the limitations as stated in **claim 12.**

1. Intel also discloses *the apparatus of claim 12, wherein: the one or more predication-support micro-operations include a first conditional move micro-operation for*

a first destination register indicated by the parallel compare instruction; the one or more predication-support micro-operations also include a second conditional move micro-operation for a second destination register indicated by the parallel compare instruction; the first proposed output value for the first conditional move micro-operation is the incoming value of the associated destination register; the second proposed output value for the first conditional move micro-operation is a first constant; the first proposed output value for the second conditional move micro-operation is the incoming value of the associated destination register; the second proposed output value for second first conditional move micro-operation is a second constant [see Intel, §4.3.1, lines 1-4; (Examiner's note: In this citation, Intel discloses the basic concept of predication wherein, depending on a predicate value, either an instruction is passed or a constant value (NOP) is passed, thus illustrating the concept of passing an argument or a constant); §4.3.3, lines 16-19 (In this citation, Intel discloses a DeMorgan-type compare function which essentially gates using two predicates, thus, in conjunction with the previous cite illustrates the use of two conditional move instructions passing either an instruction or a constant].

2. The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the

art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 14**, Morrison and Intel disclose the limitations as stated in **claim 13**.

Intel further discloses *the instruction* [being] *an "and.orcm" form of a parallel compare instruction* [see Intel, Table 4-9, Compare Type: DeMorgan, Completer: and.orcm].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 15**, Morrison and Intel disclose the limitations as stated in **claim 13**.

Intel further discloses *the instruction* [being] *an "or.andcm" form of a parallel compare instruction* [see Intel, Table 4-9, Compare Type: DeMorgan, Completer: or.andcm].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 27**, Morrison discloses the limitations as stated in **claim 26**.

Morrison does not explicitly disclose *if the instruction indicates a known constant as a first potential output value, generating a micro-operation that receives a second potential output value as an input, wherein execution of the micro-operation is to cause a processor to move either the constant value or a second potential output value into a destination register* [see Intel, §4.3.1, lines 1-4; Examiner's note: In this citation, Intel discloses outputting either an instruction or a NOP (no instruction, which is viewed as a constant value).].

Regarding **claim 33**, Morrison discloses the limitations as stated in **claim 31**.

Morrison does not explicitly disclose *the predicated instruction* [being] *a compare instruction.*

Intel does disclose *the predicated instruction* [being] *a compare instruction.*

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 41**, Morrison discloses the limitations as stated in **claim 39**.

Morrison does not explicitly disclose *the first proposed output value is the incoming value of the associated destination register; and the second proposed output value is a constant value.*

Intel does disclose *the first proposed output value is the incoming value of the associated destination register; and the second proposed output value is a constant value* [see Intel, §4.3.1, lines 1-4; Examiner's note: In this citation, Intel discloses the first value to be an instruction and the second to be the lack of instruction (NOP), thus a constant value.]

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 42**, Morrison and Intel disclose the limitations as stated in **claim 33**.

Intel further discloses *the instruction* [being] *an "or" form compare instruction* [see Intel, Table 4-9, Compare Type: OR, Completer: "or"].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 43**, Morrison and Intel disclose the limitations as stated in **claim 33**.

Intel further discloses *the instruction* [being] *an "and" form compare instruction* [see Intel, Table 4-9, Compare Type: AND, Completer: "and"].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 44**, Morrison and Intel disclose the limitations as stated in **claim 33**.

Intel further discloses *the instruction* [being] *an "and"-complement form compare instruction* [see Intel, Table 4-9, Compare Type: AND, Completer: "andcm"].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 45**, Morrison and Intel disclose the limitations as stated in **claim 33**.

Intel further discloses *the instruction* [being] *an "or"-complement form compare instruction* [see Intel, Table 4-9, Compare Type: OR, Completer: "orcm"].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 46**, Morrison discloses the limitations as stated in **claim 31**.

Morrison does not explicitly disclose *the instruction* [being] *a parallel compare instruction.*

Intel does disclose *the instruction* [being] *a parallel compare instruction* [see Intel, §4.3.3, lines 13-14].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 47**, Morrison and Intel disclose the limitations as stated in **claim 38.**

Intel also discloses *the apparatus of claim 12, wherein: the one or more predication-support micro-operations include a first conditional move micro-operation for a first destination register indicated by the parallel compare instruction; the one or more predication-support micro-operations also include a second conditional move micro-operation for a second destination register indicated by the parallel compare instruction; the first proposed output value for the first conditional move micro-operation is the incoming value of the associated destination register; the second proposed output value for the first conditional move micro-operation is a first constant; the first proposed output value for the second conditional move micro-operation is the incoming value of the associated destination register; the second proposed output value for second first conditional move micro-operation is a second constant* [see Intel, §4.3.1, lines 1-4; (Examiner's note: In this citation, Intel discloses the basic concept of predication wherein, depending on a predicate value, either an instruction is passed or a constant value (NOP) is passed, thus illustrating the concept of passing an argument or a constant); §4.3.3, lines 16-19 (In this citation, Intel discloses a DeMorgan-type compare function which essentially gates using two predicates, thus, in conjunction with the previous cite illustrates the use of two conditional move instructions passing either an instruction or a constant].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 48**, Morrison and Intel disclose the limitations as stated in **claim 46**.

Intel further discloses *the instruction* [being] *an "and.orcm" form of a parallel compare instruction* [see Intel, Table 4-9, Compare Type: DeMorgan, Completer: and.orcm].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention,

such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 49**, Morrison and Intel disclose the limitations as stated in **claim 46**.

Intel further discloses *the instruction* [being] *an "or.andcm" form of a parallel compare instruction* [see Intel, Table 4-9, Compare Type: DeMorgan, Completer: or.andcm].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

10. Claims 16-18, 23, 29, 50-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morrison in view of Dulong ("The IA-64 Architecture at Work", Carol Dulong, Intel Corp., 1998).

Regarding **claim 16**, Morrison discloses the limitations as stated in **claim 3**.

Morrison does not explicitly discloses *the one or more predication support micro-operations includes an append operation to append a qualifying predicate value to a proposed output value*.

Dulong does disclose *the one or more predication support micro-operations includes an append operation to append a qualifying predicate value to a proposed output value* [see Dulong, Page 1, Column 1, lines 30-32].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 17**, Morrison discloses the limitations as stated in **claim 3**.

Morrison does not explicitly disclose *the instruction* [being] *a load instruction; and the one or more predication support micro-operations* [including] *an append operation to append a qualifying predicate value to a load address.*

Dulong does disclose *the instruction* [being] *a load instruction; and the one or more predication support micro-operations* [including] *an append operation to append a* *qualifying predicate value to a load address* [see Dulong, Page 1, Column 1, lines 30-32; Examiner's note: Dulong discloses appending a predicate to *all* instructions, inherently including a load instruction.].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 18**, Morrison discloses the limitations as stated in **claim 3**.

Morrison does not explicitly disclose *the instruction* [being] *a store instruction; and the one or more predication support micro-operations* [including] *an append operation to append a qualifying predicate value to a store address.*

Dulong does disclose *the instruction* [being] *a store instruction; and the one or more predication support micro-operations* [including] *an append operation to append a qualifying predicate value to a store address* [see Dulong, Page 1, Column 1, lines 30-32; Examiner's note: Dulong discloses appending a predicate to *all* instructions, inherently including a load instruction.].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 23**, Morrison discloses the limitations as stated in **claim 22 (21)**.

Morrison does not explicitly disclose *a micro-operation to store an incoming value of the destination register is further to append a qualifying predicate value for the instruction with the stored incoming value.*

Dulong does disclose *a micro-operation to store an incoming value of the destination register is further to append a qualifying predicate value for the instruction with the stored incoming value* [see Dulong, Page 1, Column 1, lines 30-32].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention,

such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 29**, Morrison discloses the limitations as stated in **claim 25**.

Morrison does not explicitly disclose *generating an append micro-operation for each destination register indicated by the instruction, wherein execution of the micro-operation is to cause a processor to append a qualifying predicate for the instruction with a potential output value, and is to further cause the appended value to be stored in a register.*

Dulong does disclose *generating an append micro-operation for each destination register indicated by the instruction, wherein execution of the micro-operation is to cause a processor to append a qualifying predicate for the instruction with a potential output value, and is to further cause the appended value to be stored in a register* [see Dulong, Page 1, Col. 1, lines 30-32].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 50**, Morrison discloses the limitations as stated in **claim 38**.

Morrison does not explicitly disclose *the instruction* [being] *a load instruction; and the one or more predication support micro-operations* [including] *an append operation to append a qualifying predicate value to a load address.*

Dulong does disclose *the instruction* [being] *a load instruction; and the one or more predication support micro-operations* [including] *an append operation to append a qualifying predicate value to a load address* [see Dulong, Page 1, Column 1, lines 30-32; Examiner's note: Dulong discloses appending a predicate to *all* instructions, inherently including a load instruction.].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 51**, Morrison discloses the limitations as stated in **claim 38**.

Morrison does not explicitly disclose *the instruction* [being] *a load instruction; and the one or more predication support micro-operations* [including] *an append operation to append a qualifying predicate value to a load address.*

Dulong does disclose *the instruction* [being] *a load instruction; and the one or more predication support micro-operations* [including] *an append operation to append a qualifying predicate value to a load address* [see Dulong, Page 1, Column 1, lines 30-32; Examiner's note: Dulong discloses appending a predicate to *all* instructions, inherently including a load instruction.].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Regarding **claim 52**, Morrison discloses the limitations as stated in **claim 38**.

Morrison does not explicitly disclose *the instruction* [being] *a store instruction; and the one or more predication support micro-operations* [including] *an append operation to append a qualifying predicate value to a store address.*

Dulong does disclose *the instruction* [being] *a store instruction; and the one or more predication support micro-operations* [including] *an append operation to append a qualifying predicate value to a store address* [see Dulong, Page 1, Column 1, lines 30-32; Examiner's note: Dulong discloses appending a predicate to *all* instructions, inherently including a load instruction.].

The advantage of utilizing a instruction set similar to that of the Intel instruction set at the time of invention would have been to utilize a well-known and tested instruction set architecture which allows for predication support. This advantage would have motivated one of ordinary skill in the art at the time of invention to utilize an instruction set that would be similar in spirit to modern processors at the time invention, such as the IA-64. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize a current instruction set within the invention disclosed by Morrison.

Conclusion

11. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Vincent Lai whose telephone number is (571) 272-6749. The examiner can normally be reached on M-F 8:00-5:30 (First BiWeek Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Fritz M. Fleming can be reached on (571) 272-4145. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Vincent Lai
Examiner
Art Unit 2181

October 26, 2006

FRITZ FLEMING
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100
10/27/2006